Amendments to the Claims:

This listing of claims will replace all prior versions, and listing, of claims in the application.

1. (Currently Amended): A computer program product for execution by a server computer for dynamically generating a wrapper object, comprising:

computer code for receiving a vendor object and superclass;

computer code for performing reflection on the vendor class;

computer code for generating a wrapper class;

computer code for instantiating the wrapper class, the instantiating including generating a wrapper object as an instance of the wrapper class; and

computer code for associating the vendor object with the wrapper object, thereby enabling specific treatment of vendor objects.

2. (Original): The computer program product of claim 1 wherein the wrapper object is dynamically generated at runtime.

3. (Original): The computer program product of claim 1 wherein the superclass is one of a pre-existing JDBC, JMS, or connector class.

4. (Original): The computer program product of claim 1 wherein the superclass includes logic to handle server side tasks.

5. (Original): The computer program product of claim 1 wherein the wrapper class is generated in bytecode.

6. (Original): The computer program product of claim 5 wherein bytecode is generated for vendor methods not implemented in the superclass.

7. (Original): The computer program product of claim 5 wherein the bytecode is generated using hot code generation.

8. (Currently Amended): The computer program product of claim 1 wherein providing the wrapper object to ~~the~~ an application program, ~~allowing~~ comprises providing the application program ~~to~~ access to standard features and non-standard vendor extensions.

9. (Currently Amended): The computer program product of claim ~~1~~ 8, wherein the standard ~~extensions~~ features are J2EE ~~extensions~~ features.

10. (Currently Amended): A computer program product for execution by a server computer for processing an invocation using a dynamically generated wrapper, comprising:

computer code for receiving an invocation call by a wrapper object, the invocation call directed to a wrapped vendor object by an application program;

computer code for initiating pre-processing by the wrapper object;

computer code for calling the wrapped vendor object by the wrapper object;

computer code for receiving a result from the wrapped vendor object by the wrapper object;

computer code for initiating post-processing by the wrapper object; and

computer code for provide the result to the application program, thereby enabling specific treatment of vendor objects.

11. (Original): The computer program product of claim 10 wherein the pre-processing including calling a pre-invocation handler.

12. (Original): The computer program product of claim 10 wherein the pre-invocation handler is configured to execute server-side code.

13. (Original): The computer program product of claim 12 wherein the server-side code includes global transaction processing code.

14. (Original): The computer program product of claim 10 wherein post-processing including calling a post-invocation handler.

15. (Original): The computer program product of claim 14 wherein the post-invocation handler is configured to perform post-processing server side tasks.

16. (Original): The computer program product of claim 15 wherein the post-processing server-side tasks include global transaction management.

17. (New): The computer program product of claim 1 wherein associating the vendor object with

the wrapper object enables the vendor object to be processed in a different manner as compared with non-vendor objects.

18. (New): The computer program product of claim 10 wherein calling the wrapped vendor object by the wrapper object enables the vendor object to be processed in a different manner as compared with non-vendor objects.

19. (New): A computer program product for execution by a server computer for dynamically generating a wrapper object, comprising:

computer code for receiving a vendor object and superclass and a non-vendor object;

computer code for performing reflection on the vendor class;

computer code for generating a wrapper class;

computer code for instantiating the wrapper class, the instantiating including generating a wrapper object as an instance of the wrapper class; and

computer code for associating the vendor object with the wrapper object; wherein associating the vendor object with the wrapper object enables the vendor object to be processed in a different manner as compared with the non-vendor object.

20. (New): A computer program product for execution by a server computer for processing an invocation using a dynamically generated wrapper, comprising:

computer code for receiving an invocation call by a wrapper object, the invocation call directed to a wrapped vendor object by an application program;

computer code for initiating pre-processing by the wrapper object;

computer code for calling the wrapped vendor object by the wrapper object;

computer code for receiving a result from the wrapped vendor object by the wrapper object;

computer code for initiating post-processing by the wrapper object; and

computer code for provide the result to the application program; wherein calling the wrapped vendor object by the wrapper object enables the vendor object to be processed in a different manner as compared with a non-vendor object.